

Tecnologie Web T
30 Giugno 2014 – Compito

Tempo a disposizione: 3 ore

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

Asta.zip	file zip contenente sorgente java/class e pagine Web per punto 1
FormWeb.zip	file zip contenente sorgente java/class e file xml/xsd per punto 2
Statistica.zip	file zip contenente sorgente java/class e file di testo per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio

Studenti in debito di Tecnologie Web L-A

Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.

I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo

ESERCIZIO 1 (11 punti)

Si realizzi un'**applicazione Web** che permetta agli utenti di partecipare a un'asta (partendo da una base d'asta e con offerte al rialzo), basata su **tecnologie Java servlet e JSP**.

In particolare, l'applicazione dovrà avere una pagina iniziale **autentica** che richieda all'utente nome e cognome (considerati identificatori univoci) e l'identificatore dell'asta a cui è interessato; l'inserimento di nome-cognome deve determinare l'apertura di una sessione utente; all'interno di quella sessione un utente può partecipare a una e una sola asta; a ogni asta possono partecipare al più 10 utenti. Per semplicità si supponga che il primo utente a richiedere un identificatore di asta non ancora utilizzato, con quella azione determina anche l'inizio di quell'asta.

Una volta autenticato, l'utente deve ricevere una pagina con i dati relativi all'asta (identificatore dell'asta) e l'offerta "vincente" corrente (importo e utente offerente, ovviamente all'inizio sarà la base d'asta); tramite quella pagina l'utente avrà inoltre la possibilità di lanciare un'offerta più alta.

L'asta termina quando sono trascorsi più di 5 minuti dall'ultima offerta oppure quando l'asta è cominciata da più di un'ora. Al momento della terminazione dell'asta, tutti gli utenti partecipanti devono essere notificati con le informazioni della terminazione e dell'offerta vincente, e le loro sessioni devono essere invalidate.

Tecnologie Web T
30 Giugno 2014 – Compito

ESERCIZIO 2 (11 punti)

Si progetti una grammatica **XML Schema**, e un suo **documento XML** di esempio, per la descrizione di un **form Web di espressione di interesse scientifico a un progetto di ricerca** nel rispetto delle seguenti specifiche:

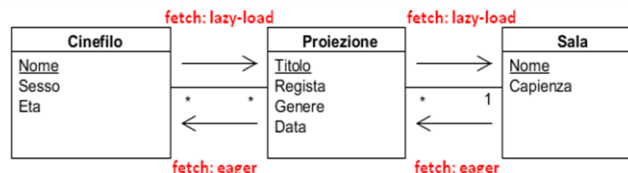
- ciascun documento XML modella un form contenente le informazioni personali del partecipante; tali informazioni sono costituite da: nome* (di tipo testuale), link a sito web personale* (di tipo indirizzo web), e mail* (di tipo email), affiliazione* (di tipo testuale), link a sito web organizzazione di appartenenza* (di tipo indirizzo web), paese*, link a fotografia personale (-) (di tipo indirizzo web), domini di interesse scientifico*, quesito su disponibilità ad ospitare un ricercatore esterno (-) e preferenza su gruppi di lavoro di interesse (-); l'area di testo relativa a "domini di interesse scientifico" ammette un insieme di keyword testuali (da un minimo di 3 a un massimo di 5 keyword); il quesito sulla disponibilità ammette una sola risposta tra {"sì", "no"}; la preferenza espressa sui gruppi di lavori di interesse rientra nell'insieme {"WG1", "WG2", "WG3"} ed è non esclusiva.

Si realizzi poi l'**applicazione Java "FormWeb"** che, facendo uso del **parser DOM**, esponga il metodo `getPartecipanti()`, unitamente a suo un `main` di prova. Nel dettaglio, tale metodo restituisce per ogni gruppo di lavoro, il numero di partecipanti (riportando per ognuno nome e affiliazione) che hanno espresso preferenza per quel gruppo, che si sono dimostrati disponibili a ospitare ricercatori esterni e nei cui domini di interesse rientrano le keyword "energy" e "big data".

N.B. Con il simbolo "*" si indica l'**obbligatorietà del campo/attributo**; con il simbolo "-" la sua **natura opzionale**.

ESERCIZIO 3 (11 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Pattern DAO** in grado di "mappare" efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean Cinefilo**, **Proiezione** e **Sala** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** dato.



Nel dettaglio, dopo aver **creato da applicazione Java** gli **schemi delle tabelle** all'interno del proprio schema nel database **TW_STUD** di **DB2** (esplicitando tutti i **vincoli opportuni**), **implementato i JavaBean** e **realizzato le classi** relative al **Pattern DAO** per l'**accesso CRUD** alle tabelle, si richiede l'implementazione di **opportuni metodi statistici per il supporto delle seguenti operazioni**:

- per ogni cinefilo minorenne, l'elenco distinto dei nomi delle sale in cui si sono tenute le proiezioni dei film a cui lo stesso ha partecipato; il numero di cinefili donna che hanno partecipato alle proiezioni tenutesi in sala "Piazza Maggiore".

Si crei poi un **main di prova** che:

- inserisca due o più proiezioni tenutesi in sale cinematografiche caratterizzate da nome (es. "Piazza Maggiore") e capienza (es. 150) e due o più spettatori che hanno preso parte alle proiezioni inserite precedentemente;
- faccia uso corretto dei metodi realizzati al punto precedente al fine di produrre una stampa (opportunamente formattata) del risultato ottenuto sul file **statistica.txt**.

N.B. L'implementazione del **Pattern DAO** deve limitarsi al solo **DBMS DB2**. La soluzione deve sfruttare i **mapping M-N** e **1-N** specificati e **propendere per il caricamento indicato nel diagramma UML**. Ogni ulteriore scelta da parte dello studente deve essere **opportunamente giustificata** con commenti nel codice.